

PATENT ABSTRACTS OF JAPAN

(11)Publication number : **2001-092985**

(43)Date of publication of application : **06.04.2001**

(51)Int.Cl.	G06T 15/00
	G06F 12/08
	G06T 1/20
	G06T 1/60
	G06T 15/50

(21)Application number : **2000-231630**

(71)Applicant : **MITSUBISHI ELECTRIC INF
TECHNOL CENTER AMERICA INC**

(22)Date of filing : **31.07.2000**

(72)Inventor : PFISTER HANSPETER
KREEGER KEVIN A
MARKS JOSEPH W
SHEN CHIA

(30)Priority

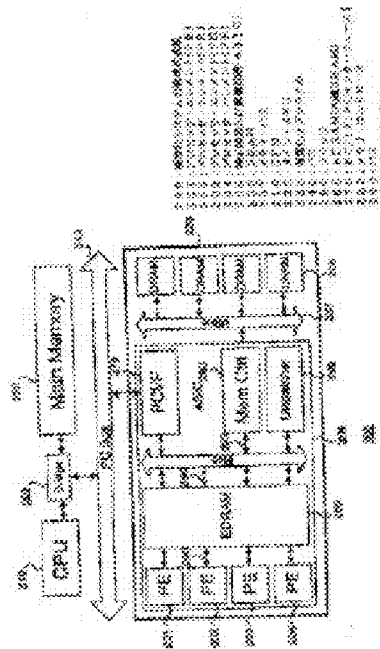
Priority number : **1999 388337** Priority date : **01.09.1999** Priority country : **US**

(54) PROGRAMMABLE VISUALIZING DEVICE FOR PROCESSING GRAPHIC DATA

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a ray tracing architecture improved for both sample value data and geometry data.

SOLUTION: This device is provided with a central processing unit(CPU) for executing a visualizing application and a scheduler. The memory of a third level is connected with the CPU and stores graphic data. The graphic data are divided into plural blocks. The memory of a second level is connected with the CPU by a system bus and stores the subsets of plural blocks. The list of blocks located in order is stored in the memory of the first level by the scheduler while being connected with the memory of the second level by a memory bus. A processor element is connected to the memory of the first level by a processor bus and a dispatcher is connected with the first or third memory and the processor element. The dispatcher transfers the blocks between the memories of the third and second levels according to the order of the block list.



【特許請求の範囲】

【請求項 1】 グラフィック・データを処理するためのプログラム可能な視覚化装置であって、視覚化アプリケーションとスケジューラとを実行するための中央処理装置と、中央処理装置に接続され複数のブロックに分割された前記グラフィック・データを記憶する前記第 3 のレベルのメモリと、システム・バスによって前記中央処理装置に接続され複数のブロックのサブセットを記憶する第 2 のレベルのメモリと、メモリ・バスによって前記第 2 のレベルのメモリと接続された第 1 のレベルのメモリであって、順に並べられたブロックのリストが前記スケジューラによって記憶される前記第 1 のレベルのメモリと、プロセッサ・バスによって前記第 1 のレベルのメモリと接続したプロセッサ・エレメントと、前記第 1、前記第 2、前記第 3 のレベルのメモリおよび前記プロセッサ・エレメントと接続され、前記リストのブロックの順序に従って前記第 3 のレベルのメモリから前記第 2 のレベルのメモリへおよび前記第 2 のレベルのメモリから前記第 3 のレベルのメモリへブロックを転送するためのディスパッチャと、を備えたことを特徴とする装置。

【請求項 2】 前記第 1 のレベルのメモリと、前記プロセッサ・エレメントと、前記ディスパッチャとが特定用途向け集積回路として構成されることを特徴とする請求項 1 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 3】 前記特定用途向け集積回路がシステム・バス・インターフェースとメモリコントローラとを含むことを特徴とする請求項 2 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 4】 前記特定用途向け集積回路と前記第 2 のレベルのメモリとがシステム・バスに接続される基板上に構成されることを特徴とする請求項 1 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 5】 前記グラフィック・データがサンプルされたデータとジオメトリ・データであることを特徴とする請求項 1 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 6】 前記サンプル・データがボリューム・サンプルであることを特徴とする請求項 5 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 7】 前記サンプル・データが画像サンプルであることを特徴とする請求項 5 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 8】 前記リスト中の前記ブロックの順序が時

間的かつ空間的であることを特徴とする請求項 1 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 9】 複数のブロックの依存状態グラフを生成してブロックの空間的および時間的順序を決定する手段をさらに備えたことを特徴とする請求項 7 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 10】 前記視覚化アプリケーションが、前記グラフィック・データの中を通過して光線をトレースし、該トレースされた光線によって前記ブロックの順序が決定されることを特徴とする請求項 1 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 11】 前記視覚化アプリケーションがグラフィック・データを分割することを特徴とする請求項 1 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 12】 前記プロセッサ・エレメントが前記複数のブロックを処理することを特徴とする請求項 1 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【請求項 13】 複数のプロセッサ・エレメントを含むことを特徴とする請求項 1 に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般に場面の視覚化に関し、さらに具体的には、プログラマブル・ハードウェアにおけるセグメンテーション、レイ・トレーシングおよびレンダリングのような視覚化タスクを行うためのハードウェア・システムである、グラフィック・データを処理するためのプログラム可能な視覚化装置に関する。

【0002】

【従来の技術】視覚化システムにおいては多くのグラフィック成分を処理して複雑な場面を正確に表す必要がある。見る人が関心領域に注意を集中できるように場面をセグメント化する必要がある場合もある。複雑な表面については、プログラマブル・シェーディングとテクスチャ・マップが要求され、リアルな照明のモデル化を行うためにリアルなライティングが必要とされる。従来技術によって、高品質の複雑な場面の描画にかかる時間量を低減することを目的とするいくつかの手法が開発されている。これらの手法には、淘汰(culling)、遅延評価、リオーダーリングおよびキャッシング(caching)が含まれる。

【0003】通常、これらの手法では、手近な特定の視覚化タスクに依存して、ハードウェアまたはソフトウェアによる解決策を利用することができる。ソフトウェア

による解決策は処理を行い易いがリアルタイムの視覚化タスクには役立たない。プログラマブル・ボリュームの視覚化タスクを行うための効率的なハードウェア・アーキテクチャの設計は、これに関わる複雑さに起因して極度に困難である。したがって、ほとんどのハードウェアによる解決策は特定用途向けのものである。

【0004】例えば、レイ・トレーシングは照明技術用として広く利用され、コンピュータ・グラフィックスの分野でリアルな画像が作成されてきた。レイ・トレーシングでは光線は単一の視点から生成される。光線は場面の中をずっとトレースされる。光線が場面の成分に出会うと、光線はリアルに反射し、屈折する。反射し、屈折した光線はさらに反射し、屈折しつづける場合もある。言うまでもなく、単純な場面においてさえ光線の数はい指数的に増加する。この理由によって、レイ・トレーシングはジオメトリ(例えばポリゴンやパラメトリック・パッチ(parametric patched)によって定義された場面のみに限られてきた。ボリューム・データ中のレイ・トレーシングは一般に難しい問題として認識されてきた。

【0005】ボリュームの視覚化を行うためにはより単純なレイ・キャスティングが一般に利用される。レイ・キャスティングとは反射光や屈折光のないレイ・トレーシングである。レイ・キャスティングでは、反射光と屈折光の影響は無視され、リアルな照明を出力する試みは他の手法によって処理される。しかし、比較的単純なレイ・キャスティングでもボリューム・データの視覚化を行うには依然として高価な計算費用を要する。この理由のために、従来技術による解決策によって特定目的用ボリューム・レンダリング・アーキテクチャが一般に提案されてきた。

【0006】最近、レイ・トレーシング・ジオメトリ・モデルのハードウェアによる加速が提案されている(「www.artrender.com/technology/ar250.html」のART参照)。このART設計には、次の光束をトレースする前にそれらの光束の終点までずっと光束をトレースする並列レイ・トレーシング・エンジンが含まれる。この入力場面データはホストのメイン・メモリに保存され、全てのプロセッサ・エレメントへ伝送された。シェーディング・サブシステムにはプログラマブル・コプロセッサを含み、同時にレイ・トレーシング・エンジンはASICを導入したものであった。

【0007】Gunther他は、「VIRIM:医学におけるリアルタイム・ボリューム視覚化用大規模並列プロセッサ」(第9回グラフィックス・ハードウェアに関するユーログラフィックス(Eurographics)ワークショップ会報、p.103~108、1994年)の中で、並列ハードウェアについて記載している。彼らのVIRIMアーキテクチャはハイデルベルグ(Heidelberg)レイ・キャスティング・アルゴリズムのハードウェアによる実現であった。ボリューム・データは各モジュール内で複製

された。VIRIMシステムは、4つのモジュールを用いて $256 \times 256 \times 128$ 個のボリュームについて10Hzを達成することができた。しかし各モジュールは3枚の基板を使用し、そのため合計12枚の基板を備えることになった。

【0008】Doggett他は、「PCIベースの双方向ボリューム・レンダリング用低コストメモリ・アーキテクチャ」(グラフィックス・ハードウェアに関するSIGGRAPH-ユーログラフィックス(Eurographics)ワークショップ会報、p.7~14、1999年)の中で、画像順序ボリューム・レンダリングを実現したアーキテクチャについて記載している。ボリュームはレンダリング用基板のDIMMに保存された。各サンプルはそのサンプルに必要なボクセルの近傍を2度読み込む。データのバッファリングは行われなかった。システムには光線の生成用プログラマブルDSPが含まれ、一方残りのパイプラインはFPGAまたはASICであった。

【0009】Pfister他は、「ボリュームProリアルタイム・レイ・キャスティング・システム」(SIGGRAPH99の会報、p.251~260)の中で、ボリュームの中を通過して投射される光線に沿ってサンプルを処理したASICパイプラインを用いてリアル・タイム・ボリューム・レンダリングを行うパイプライン化されたレンダリング・システムについて記載している。キューブ4は新しいメモリ・スキューイング方式を利用して隣接ボクセルへの回線競合のないアクセスを提供するものである。ボリューム・データは後で再利用するためにチップ上でFIFOキューに記憶された。

【0010】これらすべての設計によってASICパイプラインが利用され、高いフレーム率の描画を必要とする多数のボリューム・サンプルが処理された。これらのシステムのコスト・パフォーマンスは、スーパーコンピュータ、特定目的グラフィックス・システム、汎用グラフィックス用ワークステーションによる最先端のボリューム・レンダリングを上回るものである。

【0011】様々な視覚化上の問題はセグメンテーションと関係する。医学用アプリケーションでは、データの各スライスを手でセグメント化され、次いで、再構成されて3Dモデルのオブジェクトへ変換された。現在の商用ソフトウェアではセグメント・スライスに対してツールとインターフェースが与えられているが、これはまだ2Dモデルを対象とするものにすぎない。3Dの結果の検査は、完了するのに現在数分かかるモデル構築ステップを必要とする。これではリアルタイム・レンダリングには役に立たないことは明らかである。この時間を減らすためには、手ではなく、ダイレクト3Dセグメンテーション機能とダイレクト・ボリューム・レンダリング(DVR)とを利用して、ボリューム・データに対するセグメンテーションとレンダリングを行うほうがよい。

【0012】しかし、3Dセグメンテーションは依然と

して複雑でかつ動的なので完全な自動化は不可能であるため、若干量のユーザー入力が必要とする。1つの着想として、ユーザーがセグメンテーションの方向のかじ取りを行って最終的に所望のオブジェクトを取り出すことを可能にすることにより、人間の自然で複雑な認知能力を必要とするスキルを利用しながら、セグメンテーション処理とレンダリングの高価な計算費用を要するタスクを行うためにコンピュータを利用するというものがある。

【0013】従来技術によるセグメンテーション手法では複雑なオブジェクト認識処理手順を用いる場合もあれば、所望のセグメンテーション処理を行うために連結してシーケンスに変換する低レベルの3D形態関数が提供される場合もある。この一連の低レベルの関数はセグメンテーション「処理」と呼ばれる。これらの低レベルの関数には閾値、侵食(erode)、拡張(dilate)、フラッド・フィル(flood-fill)のような形態演算が一般に含まれる。医学用セグメンテーション・システムの典型的ユーザーが直感的でかつ簡単に使用できるように上記の方法が示されてきた。ユーザーがこの処理に対する制御を行うので、ユーザーには結果に対する信頼感が与えられる。

【0014】別のシステムでは、セグメンティングを行っている間双方向のフィードバックがユーザーに与えられる。低レベルの関数が適用された後、結果として得られるセグメント化されたボリュームがユーザーに対して表示され、ユーザーは次に行うオペレーションを選択することが可能になった。1つの演算結果によって、ユーザーは次の機能・関数の選択時に助けられた。したがって、双方向は一度に1つの低レベルの関数に限定された。ユーザーは長いシーケンスのステップを作成してある一定のセグメンテーション上の問題を処理し、シーケンスの最中で低レベルの関数のうちの1つに対してパラメータを変化させた効果を見たい場合、フィードバックは3Dインタラクティブにならない。代わりに、ユーザーはその処理で繰り返し各ステップの中を一步步に進み、毎回パラメータを変化させざるを得なかった。さらに、汎用プロセッサの使用に起因して、これらの機能を行うのに要する時間は5乃至90秒、そしてその結果の描画にプラス10秒要していた。

【0015】ある代替システムでは、現在表示されているボリュームの3つの直交スライスに対してセグメンテーションだけを行うこともできる。セグメンテーションは3つの2Dスライスに限定されていたので、毎回最初からセグメンテーション「処理」全体を行うことができた。したがってユーザーは、関数用パラメータの調整を行う制御をスライドさせながら、双方向のフィードバックを行うことができた。遺憾ながら、ボリュームの3D投影は完了するのに数分もかかることがあった。さらに、3次元では領域が大きくなり、元のスライスへ戻る

場合もあったので、2D連結成分の処理を行う同様のアプローチは存在しなかった。したがって連結された成分の処理は低速のフィードバックに限定された。

【0016】最近、同じ低レベルの機能のシーケンス処理を行う分散処理環境が提案された。この解決策によって、ボリューム・セグメンテーションにおける高レベルのデータの並列処理が認められ、DECmpp12000大規模並列プロセッサを利用してこの並列処理が行われた。DECmppはPEのMasPar SIMDメッシュの装置(導入)である。このプロセッサのパフォーマンスが非常に小さな 96^3 のボリュームのサンプル・データについて測定された。Mooreの法則による今日の技術から高速のクロックレートと 256^3 のボリュームに対してこのパフォーマンスの外挿を行うには1.14秒の時間を要するであろう。

【0017】最近、CM-200 SIMD大規模並列プロセッサで低レベルのセグメンテーション演算を行うことが提案された。 128^2 個のPEのメッシュを利用することにより、 256^3 のボリュームに対して1回の演算当たり0.6~1.3秒の間で各低レベルの演算を行うことが可能であった。今日のより高速な技術をもってしても、少数の低レベル・タスクを処理するのでさえ1回の演算当たり0.3~0.65秒の非双方向フレーム・レートを必要とする。

【0018】キャッシングによるデータ・コヒーレンスの利用は、コンピュータ・グラフィックにおいて効率を高めるもう1つの周知の手法である(Sutherland他著「10の隠された面のアルゴリズムの特徴づけ」、コンピューティング・サーベイ6(1)、p.1~55、1974年3月参照)。計算のコヒーレンスの増加によって使用メモリ量と、計算に必要な時間およびこれら双方の低減が可能となる。レイ・トレーシングを用いるシステムでは、場面の中を進む光線のコヒーレンスが走査光線ツリーによって増加し、光線を処理してコヒーレント束の中へ入れることがある。

【0019】同様に、共通の起源を持つ光線を集めて円錐台に変えることができる。これによってオブジェクトとの交差を見つけ出す時間が減少する。画像平面にわたるスペース・フィリング・カーブを用いて光線をリオーダーし、深さ優先レイ・トレーサで放出光線(spawned ray)のコヒーレンスの改善を図ることができる。モンテカルロ(Monte Carlo)レイ・トレーシング・システムの設計では、プロセッサ・キャッシュからディスク保存までのすべてのメモリ階層レベルにわたってコヒーレンスの改善が行われるようになっている。

【0020】Pharr他は、「メモリ・コヒーレント・レイ・トレーシングを用いた複雑な場面のレンダリング」(SIGGRAPH 97の会報、p.101~108)の中で、キャッシュされたレイ・トレーシング・システムについて記載している。このシステムでは、テクスチ

ャ・タイル、場面ジオメトリ、待機光線および画像サンプルがディスクに保存された。カメラによって生成された光線は分割されてグループになった。メイン・メモリに保存された場面に依存して、また、光線の処理がレンダリング作業を進める程度に依存して、このグループの光線の処理を行うスケジュールが立てられた。スケジュールされた光線は並んでメイン・メモリに記憶された。場面ジオメトリは必要なものとしてメイン・メモリに加えられた。レイ・トレーシング中生成された新しい光線が待機光線のキューに加えられた。本質的に、このシステムは、2つのレベルのキャッシュ(ディスクとメイン・メモリ)を持つメモリ階層および単一プロセッサを有するものと考えられる。これは、基本的に、キャッシング問題を解決するためのソフトウェアによる解決策である。また、Pharr他は、その時点における単一の画像について論じているにすぎず、彼らのコヒーレンシー・アルゴリズムは単に空間的ローカリティに関係するものにすぎない。

【0021】上記システムの設計は、単一タイプのジオメトリ・プリミティブのみを処理することによりある一定の利点を得るようにするためのものであった。「我々のレイ・トレーサの弁別的特徴は単一タイプのジオメトリ・プリミティブすなわち三角形がキャッシュされることである。この事実にはいくつかの利点がある。単一のケースについて光線の交差テストを最適化することができ、異なるタイプのプリミティブを保存するのに必要なスペース量の変動が少なくなるので、ジオメトリ・キャッシュのメモリ管理が容易になる。ただ1つのタイプのプリミティブだけをサポートする場合、描画装置の多くの他の部分を最適化することも可能となる。レイ(REYES)アルゴリズムは単一の内部プリミティブ(マイクロポリゴン)を同様に用いて、シェーディングとサンプリングをより効率的なものにするものである。レイ(REYES)アルゴリズムとは異なり、本発明ではシステムが最適化され大きなデータベースの三角形が処理される。これによってシステムが、走査されたデータ、科学的データおよびモザイク模様のパッチを含む、多種多様のジオメトリの共通ソースを効率的に処理することが可能となる。この単一表現の潜在的欠点は、球のような他のタイプのプリミティブがモザイク模様にされた後に多くの保存用スペースを必要とするという点である。しかしながら、単一表現の利点の方がこの欠点をしのぐことが判明した。」(前掲書p.102)。

【0022】彼らのジオメトリ・キャッシュは、いわゆる「ボクセル」すなわち三角形を取り囲むジオメトリ・グリッドの中で組織化された。レイ・トレーシングでは、「ボクセル」という用語はボリューム・レンダリングにおける意味とはまったく異なる意味を持つことに注意されたい。ボリューム・レンダリングでは、ボクセルとは3次元(ボリューム)データ・セット中の単一サンプ

ルである。以下の説明でこれらの全く異なる意味を区別するために、「ボクセル」という用語は常にボリューム・サンプルを意味し、「ブロック」という用語はキャッシュの粒状性(granularity)を指すものとする。Pharr他はブロック・サイズされた量で三角形をキャッシュした。1ブロック当たり数千個の三角形によって、キャッシングを行うのに好適なレベルの粒状性が生み出された。しかし、さらに細かな粒状性を求めて数千個の三角形を収める加速グリッドも使用された。

【0023】処理対象ブロックのスケジュールを行う目的で、各ブロックについてコスト値と利益値とが関連づけられた。コストはブロックの処理計算上の複雑さに基づき、利益は計算の完了へ向けてどれだけの進行が達成されるかによって推定された。彼らのスケジューラはこれらの値を用いて最高の対コスト利益率を持つブロックを選択することにより処理対象ブロックの選択を行った。

【0024】三角形のような2つ以上の単一グラフィック・プリミティブで表現される場面の描画をレイ・トレーシングを用いて行うことが望ましい。さらに、ソフトウェアとハードウェア・キャッシュを用いることにより追加的パフォーマンスの改善を図ることが望ましい。さらに、単純なコスト対利益アルゴリズム以上にブロック・スケジューリングを改善することが望ましい。また、画像のシーケンスすなわちフレームを描画し、空間的コヒーレンスに加えて時間的コヒーレンスを提供することも望まれる。さらにプログラマブル・ハードウェア・アーキテクチャを設けて複雑な視覚化タスクを行うことが望ましい。

【0025】

【発明が解決しようとする課題】サンプル値データとジオメトリ・データの双方のための改善されたレイ・トレーシング・アーキテクチャを提供することが本発明の目的である。このサンプル値データは2D、3Dあるいはさらに高次のサンプル値データであってもよい。ジオメトリ・データはポリゴン、パラメトリック・パッチあるいは分析的に定義されたデータであってもよい。階層メモリに埋め込み型DRAM技術を提供して、リアルタイム・レンダリング・レートを達成することがもう1つの目的である。複数のレベルのメモリ・コヒーレンシーを用いて1桁大きいパフォーマンスの改善を図ることがさらなる目的である。セグメンテーション、レイ・トレーシング・レンダリングおよびその他のグラフィック処理をサポートするプログラム可能な視覚化エンジンを提供することもまた1つの目的である。

【0026】さらに具体的には、本発明は、グラフィック・データを処理するためのプログラム可能な視覚化装置を提供するものである。該装置には視覚化アプリケーションを実行するための中央処理装置と、スケジューラとが含まれる。第3のレベルのメモリは中央処理装置と接続している。第3のレベルのメモリはグラフィック・デ

ータを記憶する。グラフィック・データは分割されて複数のブロックになる。第2のレベルのメモリはシステム・バスによって中央処理装置と接続している。第2のレベルのメモリは複数のブロックのサブセットを記憶する。第1のレベルのメモリはメモリ・バスにより第2のレベルのメモリと接続している。

【0027】スケジューラは第1のレベルのメモリの中にブロックの順序付けられたリストを記憶する。プロセッサ・エレメントはプロセッサ・バスによって第1のレベルのメモリと接続している。ディスパッチャは第1、第2、第3のメモリおよびプロセッサ・エレメントと接続している。該ディスパッチャは、ブロックのリストの順序に従って、第3のレベルのメモリから第2のレベルのメモリへ、および、第2のレベルのメモリから第3のレベルのメモリへブロックを転送する。

【0028】さらに具体的には、ある方法によってグラフィック・データの中を光線がトレースされる。該グラフィック・データにはサンプルされたジオメトリ・データが含まれる。該方法は、スケジューリング・グリッドに従ってグラフィック・データをパーティションに分割して複数のブロックにする。各ブロックについてレイ・キューが生成される。レイ・キューの各エントリはブロックの中でトレースされる光線を表す。これらのレイ・キューは、依存状態グラフを用いて空間的および時間的順序で並べられる。この順に並べられたリストに従って光線はブロックの中を通過してトレースされる。

【0029】

【課題を解決するための手段】上記の目的に鑑み、この発明は、グラフィック・データを処理するためのプログラム可能な視覚化装置であって、視覚化アプリケーションとスケジューラとを実行するための中央処理装置と、中央処理装置に接続され複数のブロックに分割された前記グラフィック・データを記憶する前記第3のレベルのメモリと、システム・バスによって前記中央処理装置に接続され複数のブロックのサブセットを記憶する第2のレベルのメモリと、メモリ・バスによって前記第2のレベルのメモリと接続された第1のレベルのメモリであって、順に並べられたブロックのリストが前記スケジューラによって記憶される前記第1のレベルのメモリと、プロセッサ・バスによって前記第1のレベルのメモリと接続したプロセッサ・エレメントと、前記第1、前記第2、前記第3のレベルのメモリおよび前記プロセッサ・エレメントと接続され、前記リストのブロックの順序に従って前記第3のレベルのメモリから前記第2のレベルのメモリへおよび前記第2のレベルのメモリから前記第3のレベルのメモリへブロックを転送するためのディスパッチャと、を備えたことを特徴とする装置にある。

【0030】また、前記第1のレベルのメモリと、前記プロセッサ・エレメントと、前記ディスパッチャとが特定用途向け集積回路として構成されることを特徴とする

請求項1に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0031】また、前記特定用途向け集積回路がシステム・バス・インターフェースとメモリコントローラとを含むことを特徴とする請求項2に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0032】また、前記特定用途向け集積回路と前記第2のレベルのメモリとがシステム・バスに接続される基板上に構成されることを特徴とする請求項1に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0033】また、前記グラフィック・データがサンプルされたデータとジオメトリ・データであることを特徴とする請求項1に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0034】また、前記サンプル・データがボリューム・サンプルであることを特徴とする請求項5に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0035】また、前記サンプル・データが画像サンプルであることを特徴とする請求項5に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0036】また、前記リスト中の前記ブロックの順序が時間的かつ空間的であることを特徴とする請求項1に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0037】また、複数のブロックの依存状態グラフを生成してブロックの空間的および時間的順序を決定する手段をさらに備えたことを特徴とする請求項7に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0038】また、前記視覚化アプリケーションが、前記グラフィック・データの中を通過して光線をトレースし、該トレースされた光線によって前記ブロックの順序が決定されることを特徴とする請求項1に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0039】また、前記視覚化アプリケーションがグラフィック・データを分割することを特徴とする請求項1に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0040】また、前記プロセッサ・エレメントが前記複数のブロックを処理することを特徴とする請求項1に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

【0041】また、複数のプロセッサ・エレメントを含むことを特徴とする請求項1に記載のグラフィック・データを処理するためのプログラム可能な視覚化装置にある。

る。

【0042】

【発明の実施の形態】論理アーキテクチャ

図1はサンプル値データ(ポリウムまたは画像)とジオメトリ・データ(ポリゴンとパッチ)の視覚化を行うための本発明によるシステム100を図示するものである。論理的には、システム100はメモリと帯域幅の階層を用いて、オンチップまたはオンボードメモリにぴったり合うサイズ以上にデータ・サイズが増加する影響を低減するようにするものである。該システムには、複数のプロセッサ・エレメント(PE)101~103、第1のレベルのキャッシュ110、第2のレベルのキャッシュ120および第3のレベルのキャッシュ130が含まれる。キャッシュとプロセッサ・エレメントはバス140によって相互に接続される。

【0043】図1ではトップからボトムへメモリ階層が小さくなり、一方帯域幅は大きくなる。したがって、各レベルは次に高いレベルのキャッシュとして機能する。データ・セット全体はメモリ階層の最低レベルまたは中間レベルにさえぴったり合う必要はない。しかし、最終画像に貢献するデータ・セットの部分がフレームからフレームへ、すなわち時間の経過と共に所定のレベルの中へぴったり合う場合、そのデータ・セットにアクセスしながら、PEはより高い帯域幅を利用することができる。

【0044】物理的アーキテクチャ

図2は本発明による視覚化システムの物理的表現200を示す図である。このシステムには、バス240によって接続される、CPU(中央処理装置)210、メイン・メモリ220および視覚化サブシステム230が含まれる。このシステムはまたCPU、メイン・メモリおよびバスを相互に接続するブリッジ250を含むことができる。

【0045】本発明によれば、このサブシステムは「プラグイン」ボードとして構成される。視覚化サブシステム230には、メモリ・バス207によってDRAM206と接続した特定用途向け集積回路(ASIC)205が含まれる。ASIC205には、バス290によって接続した、プロセッサ・エレメント(PE)201~204、内蔵DRAM(eDRAM)260、PCインターフェース(PC/IF)270、メモリ・コントローラ280およびディスプレイパッチャ292が含まれる。1つの実施の形態では、ASICによってコヒーレント・レイ・トレーシングが行われる。他のアプリケーションにはセグメンテーションとモデリングが含まれる場合もある。

【0046】CPU210は、標準的マイクロプロセッサ(例えばインテル社製ペンティアム[™]・チップ)であればいずれのものであってもよい。メイン・メモリ220はSIMMまたはDIMMから成るものであってもよい。視覚化サブシステムの基板230はPCシステム・

バス240(このバスは例えばPCI、AGP、あるいはSGIのIVCであってもよい)にプラグインすることができる。DRAM206はSDRAMまたはダイレクトRamBUSであってもよい。ASIC205には、少なくとも1つの処理ユニット並びに少なくとも1つのバンクの中に組織化されたeDRAMメモリが含まれる。例えば、三菱電機製のM32チップを使用してもよい。M32には単一のPEが含まれる。最新のeDRAM技術を備えたASICによって複数のPEが可能になり、従来のものよりずっと大きなオンチップメモリを持つことが可能になる。プロセッサ・エレメントはチップ内部のローカルな内蔵メモリと通信を行うとき非常に高い帯域幅を有する。1つのASIC当たり複数のPEを備えた複数のASICによって基板に対してさらに大きなパフォーマンスが生み出される。このシステム・アーキテクチャは、eDRAMデバイスに現在保存されたデータを迅速にPEへ転送するこの帯域幅を利用するものである。

【0047】通常のサイズのデータ・セットはeDRAMの範囲にぴったり合うことはない。したがって、外部DRAMメモリ206を利用してASICの内のプロセッサ・エレメントでデータを利用する準備ができるようになるまで、データを保存することができる。DRAMの範囲にぴったり合わない非常に大きなデータ・セットについては、サブシステムでデータの処理を行う準備ができるようになるまでメイン・メモリを利用する。これによって、データ参照のローカルリティを最適化する論理設計から3段のメモリ階層が形成される。

【0048】メモリ・コヒーレンス

レイ・トレーシング・アプリケーションでは、サブシステムによって処理の対象となるデータのメモリ・コヒーレンシーが最適化される。ブロックのメモリ領域にアクセスすることにより、また、別のブロックへ移動する前に各ブロックに対して可能な限り多くの処理を行うことによりこの最適化は行われる。本発明による処理ではブロックが順に並べられ、行うべき多量の作業があるブロックの処理を行ってから、作業量の少ないブロックの処理が行われる。また、たとえ前のブロックの方の処理作業量が少ない場合でも、後のフレームに使用されるブロックよりも先に前のフレームに使用されるブロックが並べられる。

【0049】しかし、本発明によるレイ・トレーシング・サブシステムは、従来のポリウム・レンダリング用アクセラレータで利用される場合のような厳密な処理オーダー・シーケンスに従うものではない。代わりに、行うべき作業は3次元ブロックにグループ分けされる。例えば、光線が投射されたり、トレースされたり、あるいはセグメンテーションが行われたりするブロック処理が行われる順序は、eDRAMメモリに「最も近く」てしかも最大数の結果を出力するブロックを次に処理する順

序に従う。以下に説明する発見的方法はブロック処理の順序を決定する方法である。

【0050】立体オブジェクトと埋め込み型ポリゴン・オブジェクトとの混合が存在するレンダリング・アプリケーションを実行するために、該システムでは「レイ・キュー」が利用される。セグメンテーションのようなボリューム処理アプリケーションを実行するために、別の方法(metric)を用いて空間的および時間的領域の双方におけるブロックの相対的な順序付けが行われる。

【0051】図3は、スケジューリング・グリッド302に
10 従って描画対象の3D場面300をブロック301に分割する方法を示す2Dトップ・ダウン・ビューを示す図である。データが交差するスケジューリング・グリッドのブロック間ですべてのサンプルされたジオメトリ・データ310が配分される。光線330が視点340から発出する。この光線は従来の方法で生成してもよい。レイ・キュー320は各ブロック301と関連付けられる。レイ・キューの各エントリ321は、そのブロックと現在交差する光線330のうちの1つの光線であ
20 って、スケジューリング・グリッド・ブロック302内に保存されているデータを用いて処理されるべく待機している光線を指定する。

【0052】光線を指定するデータにはその(x、y、z)位置と、次のトレース・ステップ位置のための増分(、x、y、z)と、現在蓄積されている光線のRGB値とが含まれる。

【0053】図4は、本発明によるレイ・トレーシング方法による処理とデータの流れを図示するものである。システムの中心部に、スケジューラ410と、先入れ先出し(FIFO)バッファ415を介してデータを通信する
30 ディスパッチャ420とがある。このスケジューラは、スケジューリング・グリッド302、光線リスト401などのアプリケーションについての詳細情報を持っている。リスト401の各エントリ402はレイ・キュー320のうちの1つを指す。

【0054】スケジューラは以下に説明する発見的処理を用いて、どのブロックを処理の対象とすべきか、および、どの順序で処理すべきかの決定を行う。この発見的処理は視覚化アプリケーションに特有の方法である。スケジューラは、FIFOバッファ415のディスパッチャ
40 によって処理するブロックのリストを保存する。ディスパッチャは個々のプロセッサ・エレメント201~204へブロックを転送する。またディスパッチャは、3つのレベルのメモリ階層(110、120、130)間でサンプルされたジオメトリデータ310の動きの制御を行う。このようにして、スケジューラによって処理対象ブロックが割り当てられるとき、プロセッサ・エレメント201~204の各々に対して正しいデータが利用可能になることが保証される。FIFOバッファにブロックが配置された後、スケジューラは、専用の視覚化アプリケー

ションとは独立に、可能な限りPEに接近するようにブロックの移動を試みる。

【0055】データはレベルに依存する異なるサイズの塊(chunk)でメモリのレベルの中を移動する。例えば、ある特定フレームのボリューム・データを階層的空間的下位区分に分割することができる。したがって、プロセッサ・エレメントがボリューム・データを必要とし、そのデータが最高レベルのメモリ階層にしか存在しないとき、大きな塊のボリュームは次の低いレベルへ動かされ、次いで、プロセッサ・エレメントにもっとも近い最低レベルまでずっと動かされることになる。

【0056】最初、データはメイン・メモリ220に記憶される。CPU210は最初の光線情報を生成し、レイ・キュー320をロードする。ディスパッチャは、第1のブロックが処理を行う準備ができたとき、基板230のDRAM206へデータ転送を開始する。eDRAM260の中へ直接ブロックをロードしてもよい。実際の作業では、メイン・メモリからの単一の読み込みとオンボードDRAMとASIC eDRAM双方への同時的書き込みを行うことによって上記を行うことができる。

【0057】ブロックが処理されると、現在のブロックのレイ・キューが読み込まれ、サンプルされたジオメトリ・データと光線をブロックの中で交差させる方法が決定される。光線がブロックから出て、次のスケジューリング・グリッド・ブロックに入るとき、この情報は、次のブロックのために適切なレイ・キューに配置される。レイ・キャスティングを行うために光線が完全不透明度に達した場合、情報が元のレイ・キューの中へ戻ることはない。

【0058】プロセッサ・エレメントが現在のブロック処理を完了した(すなわち、現在のブロックのキューにもはや光線が存在しない)とき、プロセッサ・エレメントはPEのための新しい処理ブロックを選択するようにスケジューラに信号を出す。スケジューラは、発見的方法に従って、レイ・キューの現在の状態およびメモリの現在の状態に基づいて最適のブロックを選択する。(レイ・キャスティング用として)すべてのレイ・キューが空になったとき、処理は完了し、表示や印刷を行うために画像サンプルを収集することができる。

【0059】スケジューリング・アルゴリズム

以下のスケジューリング・アルゴリズムを用いてブロックの処理順序を選択することができる。

【0060】MAX作業

このアルゴリズムは、例えば、処理を待機しているそのレイ・キューの中で最大数の光線を含むブロックのスケジュールを決めるものである。

【0061】MAXロード作業

このアルゴリズムは、最低レベルの(したがってプロセッサ・エレメントに最も近い)メモリ階層の中へロード

されたブロックを調べ、そのブロックのキューの中で最大数の光線を持つブロックを選択するものである。すべてのロードされたブロックが空のレイ・キューを有する(すなわち行うべき作業が存在しない)場合、たとえメモリ階層中に次善(next best)のブロックが現在記憶されていても、MAX作業アルゴリズムを用いて、スケジュール対象の次善(next best)のブロックが選択される。

【0062】ジオメトリック

このアルゴリズムは、第1の光線(すなわち、反射も屈折も存在しない)光源またはビューイング円錐台(viewing frustum)のいずれかから発する光線はすべてその発生点から外へ向かって進むという事実を利用するものである。したがって、ブロックは、光線の発生点すなわち視点340に最も近い地点から光線の発生点から最も遠い地点へという順序で処理される。したがって、そのブロックの中へ任意の光線を送ることができるすべてのブロックは、任意のブロックが処理される前に、既に処理されてしまっていることになる。このようにして各ブロックは一度で正確に処理が行われる。言うまでもなく、この処理は、放出光線が共通の発生点を持たずにほとんどランダムな方向に進む場合には、レイ・トレーシングについて当てはまらない。そのため他のアルゴリズムも依然として必要とされる。

【0063】依存状態グラフ

最適の処理対象ブロックの測定値としてレイ・キュー中の光線の数を用いる代わりに、この方法では依存状態グラフが利用される。この方法では時間にわたるフレーム間のコヒーレンスが利用される。各フレームについて依存状態グラフが生成される。この依存状態グラフは、どのブロックが他のどのブロックへ光線を送っているかを示すものである。後のフレームについては、視点(すなわちカメラ・アングル)はおそらくほんの少量しか動いていない。したがって、前のフレーム用として使用されたブロックのほとんどは次のフレームについてもまだ有効である。したがって、他のどのブロックより前にどのブロックを処理したほうがよいかを推定する1つの指針として前のフレームから作製したグラフを利用することが可能である。

【0064】図5の(a)は、視点502からスケジュールリング・グリッド503の中を通る投射光線501の一例を示す図である。前の光線がまだ終了点に達していない場合、若干の光線が5つのブロックを走査する可能性がある。図5の(b)は依存状態グラフ510を図示する。このグラフでは、ノード511はブロックであり、矢印のついたエッジ512は1つのブロックを出て次のブロックに入る光線を示している。図5の(b)ではブロックからブロックへの光線の流れを見ることは難しい。図5の(c)は図5の(b)と同じ依存状態グラフを図示するものであるが、すべての矢印のついたエッジがトップ

からボトムへ流れるようにノードが並べ替えられている。ここでは、ブロックを処理する最適順序が明瞭である。

【0065】レイ・キャストリングを用いて単純な2Dの3×3からなるスケジュールリング・グリッドについて、依存状態グラフを描いて、ブロックの最適の処理順序を決定できるようにこの依存状態グラフを並べ替えることは簡単である。しかし、システム100のスケジュールリング・グリッドは相当のサイズの3Dサンプル値データとなる可能性がある。このため手による分析を行うことは困難であるものの、計算によって最適順序を解くことは依然として可能である。

【0066】さらに、本発明によるシステムはレイ・トレーシングをもサポートするものである。図6の(a)は、あるレイ・トレーシングを行うための同じ単純な3×3からなるスケジュールリング・グリッド503を示し、この場合、オブジェクト600はグリッドの中心に配置され、さらに、これらの光線には反射光線601が含まれる。図6の(b)には、対応する依存状態グラフ610が処理ブロックの順序を示すには明瞭でないことが示されている。点線のエッジ611は、グラフでサイクルを生じる2次反射光線601に起因する依存性を示す。1つの順序では1次光線は反射光線の前に処理される。

【0067】図7の(a)と(b)にはどのようにして時間的コヒーレンスに達するかが示されている。図7の(a)では、グリッド700はグラフィック・オブジェクト701(例えば球)にスーパーインポーズされる。レイ・トレーシングに必要なブロック702は黒(ハッチング)で示されている。時刻 t_0 におけるフレームの光線703はカメラ位置704から生成される。時刻 t_1 でカメラ位置705はシフトしている。次のフレームのために多数のブロックを再利用することができる。そしてこれらのブロックは最低レベルのキャッシュの中にそのまま残る。新しいブロック707(粗いハッチング部分)だけを特定する必要がある。

【0068】置換アルゴリズム

ディスパッチャが特定のメモリの中へブロックをロードする必要があり、記憶領域がいっぱいに満たされているとき、以前に記憶された若干のブロックを除去する必要がある。以下の置換アルゴリズムを用いて除去対象ブロックの選択を行うことができる。

【0069】ランダム・アルゴリズム

これは、新しいブロックのためにスペースをあけるためのランダムな除去用ブロックの選択だけを行う単純な置換アルゴリズムである。場合によっては、このブロックはすぐに必要となるブロックになる場合もある。

【0070】MIN作業

これは、MAX作業スケジュールリング・アルゴリズムと論理的に正反対のものである。このアルゴリズムは、多

数の光線を持つブロックの前に処理対象の少数の光線を持つブロックを置き換えるものである。

【0071】依存状態グラフ

これは、スケジューリング・アルゴリズムと同様に、フレーム間コヒーレンスと、前のフレームからつくられた依存状態グラフとを利用するものである。

【0072】推奨される実施の形態を挙げて本発明について説明してきたが、本発明の精神と範囲の中で様々な適化と改変を行うことが可能であることが理解されるであろう。したがって、本発明の精神と範囲の中に入るものとしてそのようなすべての変更例および改変例をカバーすることが添付の請求項の目的である。

【0073】

【発明の効果】以上のようにこの発明によれば、グラフィック・データを処理するためのプログラム可能な視覚化装置であって、視覚化アプリケーションとスケジューラとを実行するための中央処理装置と、中央処理装置に接続され複数のブロックに分割された前記グラフィック・データを記憶する前記第3のレベルのメモリと、システム・バスによって前記中央処理装置に接続され複数のブロックのサブセットを記憶する第2のレベルのメモリと、メモリ・バスによって前記第2のレベルのメモリと接続された第1のレベルのメモリであって、順に並べられたブロックのリストが前記スケジューラによって記憶される前記第1のレベルのメモリと、プロセッサ・バスによって前記第1のレベルのメモリと接続したプロセッサ・エレメントと、前記第1、前記第2、前記第3のレベルのメモリおよび前記プロセッサ・エレメントと接続され、前記リストのブロックの順序に従って前記第3のレベルのメモリから前記第2のレベルのメモリへおよび前記第2のレベルのメモリから前記第3のレベルのメモリへブロックを転送するためのディスパッチャと、を備えたことを特徴とする装置としたので、サンプル値データとジオメトリ・データの双方のための改善されたレイ・トレーシング・アーキテクチャを提供することができる。

【図面の簡単な説明】

【図1】 本発明によるプログラム可能な視覚化システム

ムの論理表現を示すブロック図である。

【図2】 本発明による視覚化システムの物理的表現を示すブロック図である。

【図3】 描画対象場面上にスーパーインポーズされるスケジューリング・グリッドを示すブロック図である。

【図4】 本発明によるコヒーレント・スケジューリングの方法を示す流れ線図である。

【図5】 スケジューリング・グリッドを走査する光線および時間的および空間的順序で並べられた処理ブロックの依存状態を示す図である。

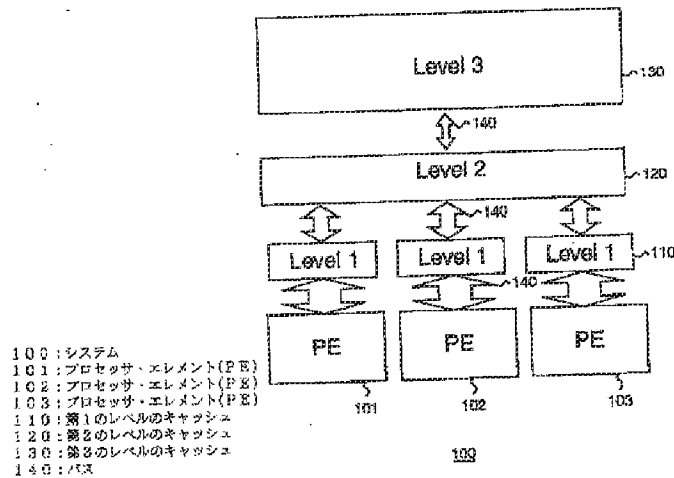
【図6】 反射光線でスケジューリング・グリッドを走査する光線および処理ブロックの依存状態を示す図である。

【図7】 時間にわたってトレースされる光線を示す図である。

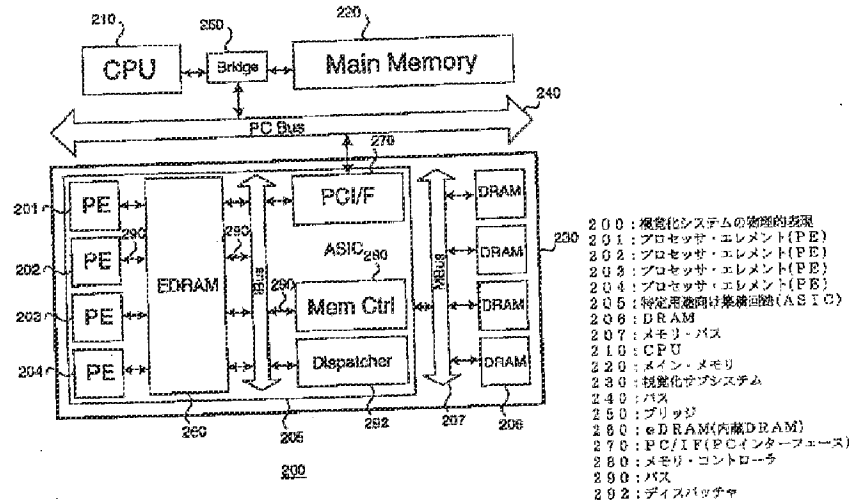
【符号の説明】

100 システム、101～103、202～204
プロセッサ・エレメント(PE)、110 第1のレベルのキャッシュ、120 第2のレベルのキャッシュ、130 第3のレベルのキャッシュ、140 バス、201 プロセッサ・エレメント(PE)、205 特定用途向け集積回路(ASIC)、206 DRAM、207 メモリ・バス、210 CPU、220 メイン・メモリ、230 視覚化サブシステム(基板)、240 バス、250ブリッジ、260 eDRAM(内蔵DRAM)、270 PC/IF(PCインターフェース)、280メモリ・コントローラ、292 ディスパッチャ、300 3D場面、301ブロック、302、503 スケジューリング・グリッド、310 ジオメトリ・データ、320 レイ・キュー、321、402 エントリ、330 光線、340 視点、401 光線リスト、410 スケジューラ、415 FIFOバッファ、420 ディスパッチャ、501 投射光線、502 視点、510、610 依存状態グラフ、511 ノード、512、611 視点のエッジ、600 オブジェクト、601 反射光線、700 グリッド、701 グラフィック・オブジェクト、704、705 カメラ位置。

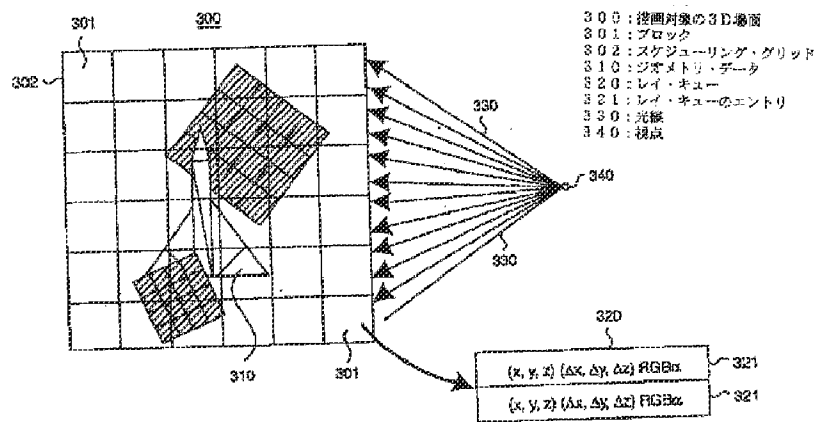
【図1】



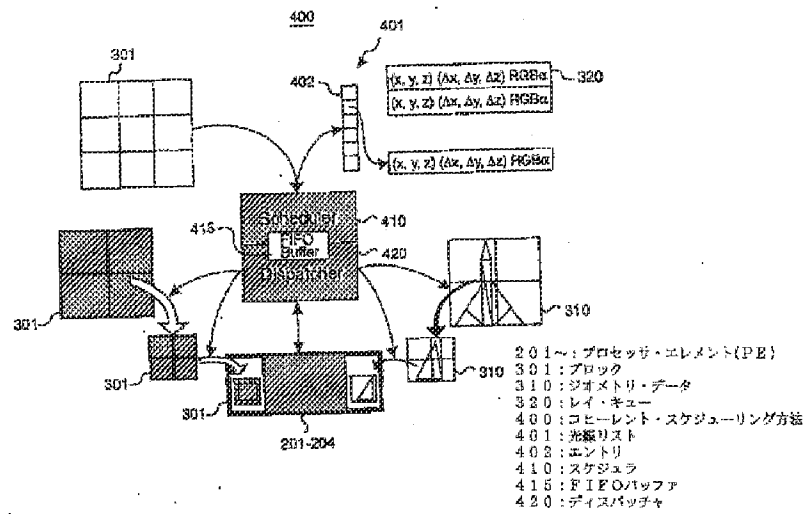
【図2】



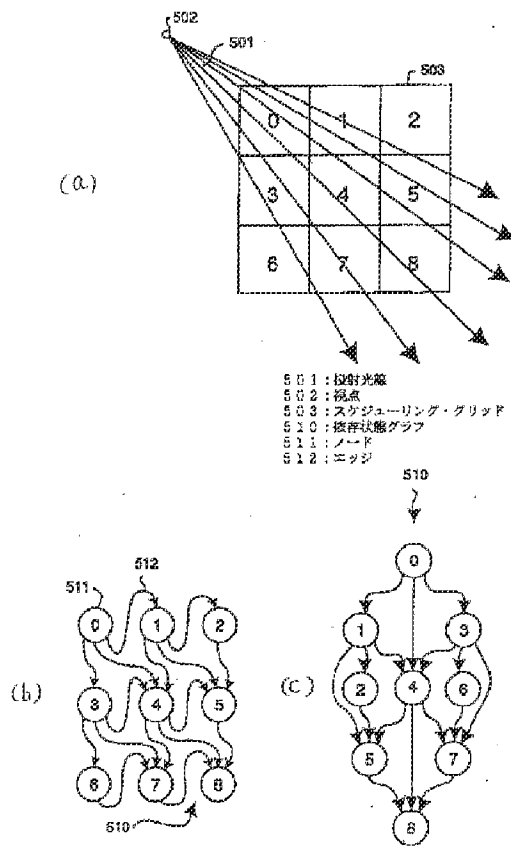
【図3】



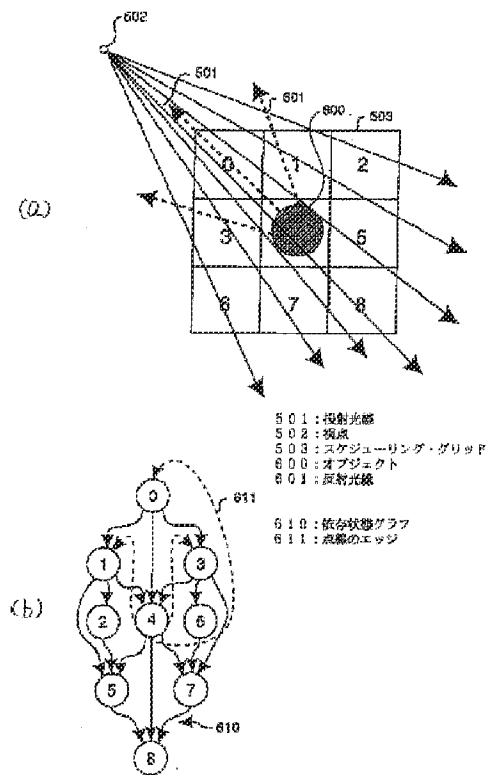
【図4】



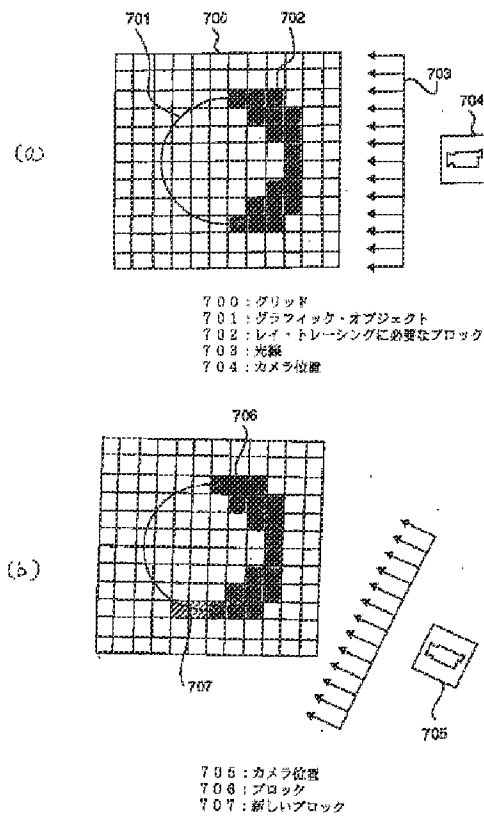
【図5】



【図6】



【図7】



フロントページの続き

(51)Int.Cl. ⁷	識別記号	F I	デマコード (参考)
G 0 6 T 15/50	2 3 0	G 0 6 T 15/50	2 3 0
(71)出願人	597067574 201 BROADWAY, CAMBRI DGE, MASSACHUSETTS 02139, U. S. A.	(72)発明者	ケビン・エイ・クリーガー アメリカ合衆国、ニューヨーク州、ポー ト・ジェファークソン、ランドール・アベニ ュー 115
(72)発明者	ハンスピーター・フィスター アメリカ合衆国、マサチューセッツ州、サ マービル、パーク・ストリート 60	(72)発明者	ジョゼフ・ダブリュ・マークス アメリカ合衆国、マサチューセッツ州、ペ ルモント、ダルトン・ロード 61
		(72)発明者	ティア・シェン アメリカ合衆国、マサチューセッツ州、ア ーリントン、アパッチ・トレイル 3